REMARKS

Claims 1-3, 14-16 and 27-29 are pending.

Claims 1-3, 14-16 and 27-29 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 5,539,907 (Srivastava).

In reply to the last Office Action, the applicants asserted that Srivastava does not teach the claimed "first compiled code state" and "second compiled code state." In the present Office Action, the Examiner continues to identify the initially compiled object modules of Srivastava as the claimed "first compiled code state" and to identify Srivastava's "single linked code module in the form of a machine independent register transfer language" as the claimed "second compiled code state." The applicants respectfully submit that Srivastava's "single linked code module in the form of a machine independent register transfer language" does not constitute the claimed "second compiled code state."

As the applicants explained in the last reply, the claimed invention is directed to a method of translating the compiled machine code of a computer program designed to run on a first processor having a first instruction set into compiled machine code designed to run on a different processor having a different instruction set. The meaning of the terms "first compiled code state" and "second compiled code state" are clear from the context in which they are used in the specification of the present application. For example, in discussing the background of the invention, the application explains:

> In certain circumstances, it is useful to translate compiled *machine code* from a first state *corresponding to a first instruction set* to a second state *corresponding to a second instruction set*. As is known, such translating is preferable to the alternative: locating the source code from which the machine code in the first code state was derived; and writing and debugging a compiler to compile such located source code directly into the *machine code* in the second code state.

Spec., pg. 1, ln. 15 – pg. 2, ln 2 (emphasis added). Thus, in the context of the present invention, the term "first compiled code state" refers to machine-specific object code corresponding to a first processor instruction set and the term "second compiled code state" refers to machine-specific object code corresponding to a different processor instruction set.

On the contrary, the system described in Srivastava has nothing to do with translating a computer program compiled to run on one processor instruction set into compiled machine code for a different processor instruction set. Rather, Srivastava describes a method that allows a developer to modify a computer program to insert certain performance monitoring code that will allow the developer to monitor the performance of the computer program when it runs on the *same* computer processor. Specifically, the object code modules of a computer program designed to run a particular computer processor are (i) translated to an intermediate form (the "single linked code module in the form of a *machine independent* register transfer language"), (ii) monitoring code is added to the program in that intermediate form, and then (iii) a code generator generates new object code including the monitoring code for execution on the *same* computer processor. There is no translation of the original object modules into object code executable on a different processor.

While the applicants submit that the meaning of the terms "first compiled code state" and "second compiled code state" are clear when read in the context of the present application, the applicants have nevertheless amended independent claims 1, 14 and 27 to further highlight this feature of the invention. Specifically, each of those claims has been amended to further recite:

> the first compiled code state comprising machine-specific object code corresponding to a first processor instruction set and the second compiled code state comprising machine-specific object code corresponding to a different processor instruction set.

See, claims 1, 14 and 27 as amended above. Srivastava does not teach or suggest translating the compiled machine code of a computer program into a "second compiled code state" where that second compiled code state "compris[es] *machine-specific object code* corresponding to *a different processor instruction set,*" as now recited in each independent claim (emphasis added). On the contrary, the "single linked code module" that the Office Action has equated with the claimed "second compiled code state" is "in the form of a *machine independent* register transfer language" (emphasis added). It is neither "machine-specific," nor does it correspond to a "different processor instruction set" as claimed. Consequently, the applicants respectfully submit that Srivastava does not teach or suggest the invention recited in independent claims 1, 14 and 27. Inasmuch as claims 2-3, 15-16 and 28-

29 depend either directly or indirectly from one of those independent claims, the applicants submit that they too are patentable for the same reasons. Withdrawal of the Section 102(e) rejection of claims 1-3, 14-16 and 27-29 is respectfully requested for this reason.

CONCLUSION

For all the foregoing reasons, the applicants respectfully submit that the present application is in condition for allowance. An early Notice of Allowance is respectfully requested.

Date: November 26, 2007

/Steven B. Samuels/
Steven B. Samuels
Registration No. 37,711

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

Correspondence address:
Richard J. Gregson, Esq.
Unisys Corporation
Unisys Way
Blue Bell, Pennsylvania 19424-0001